
# Understanding Causal Systems

Presented By: David N. Card
card@systemsandsoftware.org

***Systems and Software Consortium*** | *2214 Rock Hill Road, Herndon, VA 20170-4227*
*Phone: (703)742-8877 | FAX: (703)742-7200*
www.systemsandsoftware.org

## Agenda

- Causality
- Defects
- CMM® and CMMI® Views of Causal Analysis
- Summary



## Who Cares?

- The structure of a causal system must be understood in order to take effective action to change its performance
- Scientists and statisticians often try to avoid making judgments about causality – engineers and doctors can't!

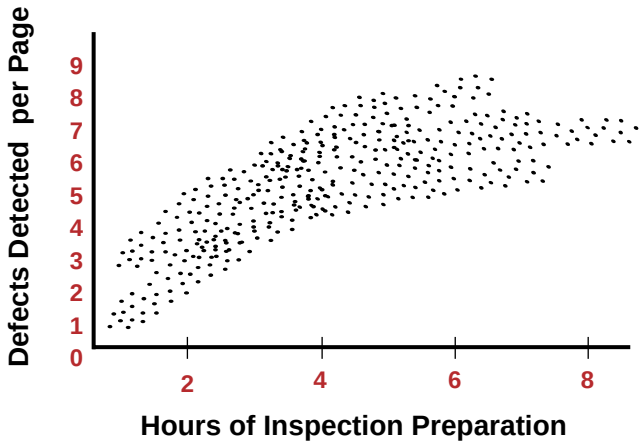

3

## Concept of Causality

- Conditions of causality
  - *Cause* and *effect* must demonstrate association
  - *Cause* must precede the *effect* in time
  - Mechanism by which the cause produces the effect must be understood
- Assignment of cause in a "human-intensive process" always includes a significant element of subjectivity


4

## A Causal Relationship?




SYSTEMS AND SOFTWARE CONSORTIUM
BUILDING BETTER SOLUTIONS TOGETHER

5

## Causal Systems

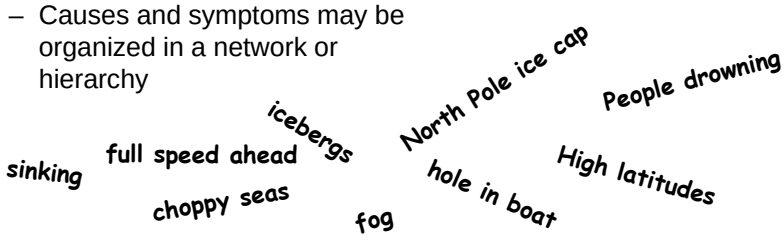
- A causal system is a network of interacting factors that affect an outcome of interest
- Causes may linked hierarchically or laterally — causes become effects
- A vocabulary limited to *cause* and *effect* is not sufficient for reasoning about causal systems

SYSTEMS AND SOFTWARE CONSORTIUM
BUILDING BETTER SOLUTIONS TOGETHER

6

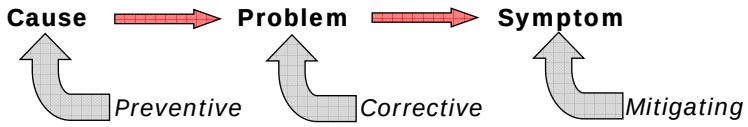
## Symptoms, Problems, and Causes

- A given problem often is associated with multiple symptoms and causes
  - Symptoms usually are the most readily visible consequences of the problem
  - Causes contribute to the occurrence of the problem
  - Causes and symptoms may be organized in a network or hierarchy




## Elements of a Causal System

***Observations***




***Actions***

- Practitioners often are not very specific about what they are observing and the nature of the action they take
- Consequently, it is hard to determine whether these are the right actions or what the likely consequence will be

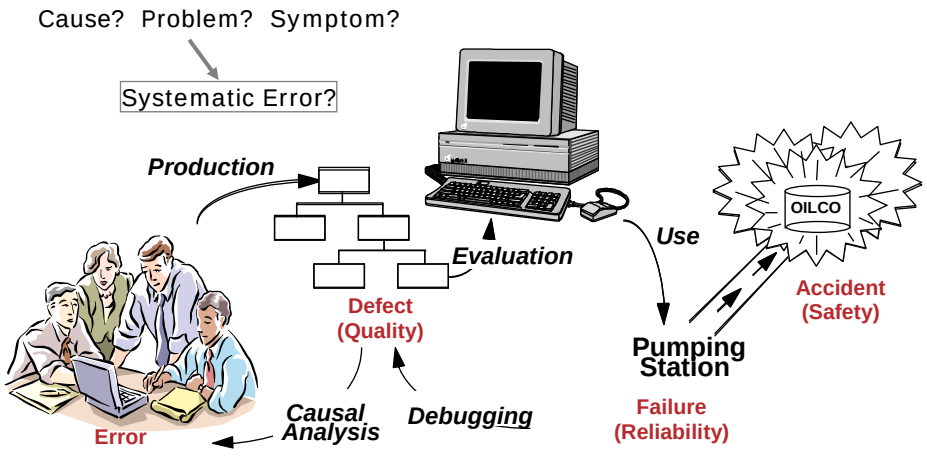
SYSTEMS AND SOFTWARE CONSORTIUM
BUILDING BETTER SOLUTIONS TOGETHER

## Concept of a Defect

- A defect is a deficiency or anomaly in an internal or delivered product that must be corrected in order to deliver the desired outcome
- The cost of rework due to defects is 30 to 50 % of the typical (e.g., CMM Level 1) project budget



## Defect Causal Chain



## Key Software Definitions

- Error: A mistake made by a member of the software team
- Defect: A section of code or documentation that must be changed to correct a failure
- Failure: A situation in which the software fails to execute as intended
- Accident: Damage to a person or property resulting from a failure
- Problem Report: A description of an undesirable outcome (e.g., error, defect, failure, or accident)

***Similar concepts may apply in systems engineering***




## What Is DCA?

- Examination of information about problems
- Intent to identify causes of defects so that they can be prevented or detected earlier, or so that appropriate corrective action can be taken
- Many different approaches, called defect causal analysis or root cause analysis, employ many different techniques
- Performed in response to an *out of control* (OOC) situation or as part of a continual improvement program
- Weak DCA leads to poor process management

## Examples of Weak Results

- Identified cause does not lead to action
  - Bad data
  - Personnel issues
- Causes and actions are superficial
  - Defect rates from inspections are low, so *reinspect*
  - Defect rates from inspections are high, so *orient* the producer
- Only a small number of problems may result in false OOC signals or OBE (overcome by events) situations
- Avoid tendency to stop at “first plausible explanation”!




## Formal DCA Process

- Pre-defined process helps to ensure “deep” rather than superficial analysis
- Typical phases:
  - Meeting/Analysis Preparation
  - Causal Analysis
  - Corrective Action Development
- Typical tools:
  - Pareto charts
  - Cause/effect (Ishikawa) diagrams
- Pre-defined process helps to ensure effective use of tools

## Relationship to CMM

- Level 4 — Defect Causal Analysis
  - May be ad hoc
  - Performed in response to *out of control* situations
- Level 5 — Defect Prevention
  - A Key Process Area (KPA) of CMM
  - Systematic approach required for DCA – “in accordance with a documented procedure”
  - Performed even when process is *in control*
  - Additional planning and feedback requirements

***CAR in CMMI is similar – more later***




## DCA for Improvement

- May be organized within a Defect Prevention context
- Assigns responsibility for causal analysis of a process to the engineering team
- Bases analysis on a sample of problems rather than an exhaustive study of all problems
- The engineering team proposes actions to
  - Prevent problems
  - Find problems earlier
- Assigns responsibility for implementing proposals to a management action team

## Defect Prevention Description

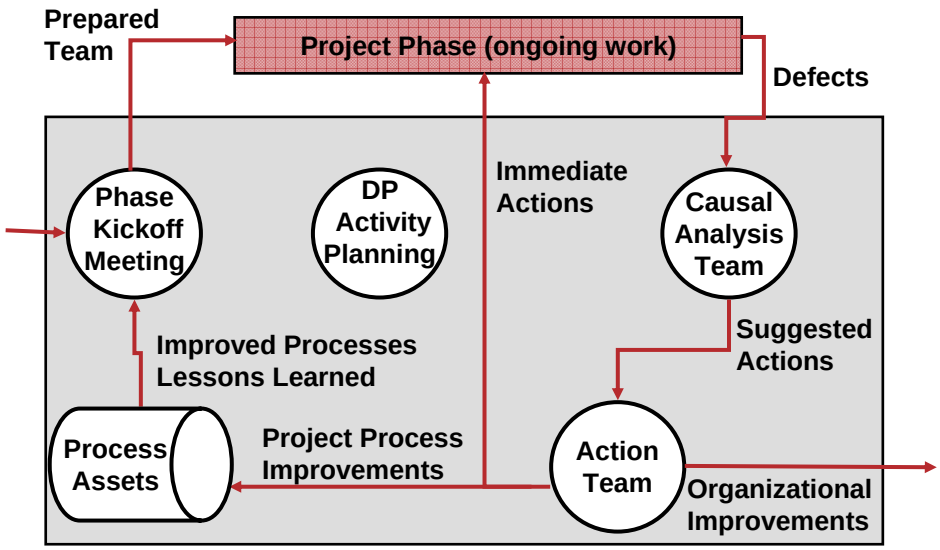
- Purpose
  - To identify the cause of defects and prevent them from recurring
- KPA goals
  - Defect prevention activities are planned
  - Common causes of defects are sought out and identified
  - Common causes of defects are prioritized and systematically eliminated

**Source:** ***Key Practices of the Capability Maturity Model, Version 1.1*****, SEI, CMU/SEI-93-TR-25.**

SYSTEMS AND SOFTWARE CONSORTIUM

17

## Project DP Process in the CMM



## DP Planning

- Based on results of process performance analysis provided by (Quantitative Process Management (QPM), Software Quality Management (SQM), Process Change Management (PCM) activities
- Defines
  - Focus of DP activities (e.g., problem area)
  - Charter, composition, roles, and responsibilities of defect causal analysis team(s)
  - Charter, composition, roles, and responsibility of action team(s)
  - Schedules for phase kickoff meetings
- May not address all project activities and products




## Phase Kickoff Meeting

- Entire project staff participates
- Typical topics
  - Lessons learned (Dos and Don'ts) from previous projects and builds
  - Defect causal analysis and other process improvement activities to be conducted
  - Goals and objectives for this phase
  - Changes to methods and tools for this phase

## Causal Analysis and Resolution

- CMMI Process Area at Level 5
- Differences from CMM DP
  - Phase Kick-off Meetings not addressed
  - Planning requirements relaxed (management versus technical plan)
  - Scope broadened to include all types of anomalies, not just defects
  - Not necessary to “prevent” defects
- DP provides the more challenging set of requirements




## Relationship to Six Sigma

- Many causal analysis techniques provided in typical Six Sigma training programs (e.g, Error Modes and Effects Analysis)
- Defect prevention planning and team-based approach to DCA (CMM requirements) usually are not explicit elements of Six Sigma
- DP in the SW-CMM, and CAR in the CMMI, assume processes are defined; the need to define processes prior to DCA increases the time and effort required

## Summary

- Basic concepts of causality are often misunderstood
- A generic model of causal systems and systematic method of analyzing them helps ensure effective actions
- DP (CMM) and CAR (CMMI) requirements differ in some important ways
- Regular and effective causal analysis is an essential element of any continuous improvement program




## Opportunity – IEEE 1044

- IEEE Standard 1044 – Classification of Software Anomalies (1995)
- Working group being established to revise this standard
- Goals of revision
  - Incorporate current concepts
    - Inspection defects
    - Orthogonal defect classification
    - Defect causal analysis
    - CMMI, Six Sigma, etc.
  - Extend to defect prevention and improvement from just problem management
- Some face-to-face meetings, but most work to be accomplished off-line

## Bibliography


- Chillargee, R., and I. Bhandari, et. al. “Orthogonal Defect Classification - A Concept for In-Process Measurements.” IEEE Transactions on Software Engineering, November 1992.
- Mays, R., et al., Experiences with Defect Prevention. IBM Systems Journal, January 1990
- Dangerfield, O., et al. “Defect Causal Analysis - A Report from the Field.” ASQC International Conference on Software Quality, October 1992.
- Yu, W. “A Software Fault Prevention Approach in Coding and Root Cause Analysi.”, Bell Labs Technical Journal, April 1998.
- Card, D. “Learning from Our Mistakes with Defect Causal Analysis.” IEEE Software, January 1998.
- Leszak, M., et al. “Classification and Evaluation of Defects in a Project Perspective.” Journal of Systems and Software, April 2002.